UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/676,802	09/30/2003	Thomas Chadzelek	09700.0054-00	3771

22852 7590 01/10/2008

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
AUGUSTINE, NICHOLAS	
ART UNIT	PAPER NUMBER
2179	
MAIL DATE	DELIVERY MODE
01/10/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/676,802	**Applicant(s)** CHADZELEK ET AL.	
	Examiner Nicholas Augustine	**Art Unit** 2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _17 October 2007_.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1 and 3-37_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1 and 3-37_ is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application

6) □ Other: _____.

DETAILED ACTION

A. This action is in response to the following communications: Request for Continued Examination filed: 10/17/2007.

B. Claims 1 and 3-37 remain pending.

C. Objections toward the specification made in previous office action are withdrawn due to amendment.

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 10/17/2007 has been entered.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

4. Claims 1 and 3-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft in view of David A. Karp et al, Windows XP In A Nutshell, April 2002, O'Reilly First Edition.

Note: Windows Explorer is the Graphical Shell used by the Microsoft Windows XP Professional Operating System.

Note: Supplemental screen captures are included with the resulting functions and action as taught by Karp.

Note: Supplemental references are provided to depict what was commonly known in the art of Windows XP as taught by Karp.

A. (http://www.delphi3000.com/articles/article_1667.asp?SK=)

B. (http://en.wikipedia.org/wiki/Container_(data_structure)#Graphic_Containers)

C. (http://en.wikipedia.org/wiki/Windows_shell)

D. (http://en.wikipedia.org/wiki/Daemon_(computer_software))

As for independent claims 1 and 18, Microsoft teaches a computer program product and corresponding method (fig.1), tangibly embodied on an information carrier(fig.2), the product comprising instructions operable to cause data processing apparatus to execute a method for navigating user interface elements on a display screen; the interface elements being arranged in order into user interface elements groups having assigned group identifier characters (identifiers within graphical containers or actual rendering of group identifiers as depicted in the figures which indicated where the controls are located in a hierarchical list); and the interface elements indicating, on the display screen, an element currently having focus to receive user input (as depicted in figure 3; focus is on a desired graphical container) detecting a user navigation input comprising one of (fig.2; e.g.- Document1 and fig.5; e.g. "3 Main"), a forward user navigation input comprises a forward modifier key press combined with a key press of a first group identifier character, and the backward user navigation input comprises a backward modifier key press combined with a key press of a second group identifier character (arrow keys, tab and shift + tab keys); identifying a selected group of user interface elements associated with the first or second group identifier character (fig.5 "child nodes"); and shifting input focus to a user interface element in the selected group based on the user navigation input ;(fig.5; wherein the user can select objects through manipulation of the keyboard, e.g. arrow keys, context menu key, shift-x, alt -x, assigned hot keys, etc...). However Microsoft does not expressly point out and describe *wherein, when the user navigation input is detected, determining a current group, that contains the interface element currently having input focus, and determining a target* *group that corresponds to the group identifier key press; wherein when the user navigation input is the forward user navigation input, input focus is shifted to an interface element next in order in the current group if the current group is the same as the target group, or input focus is shifted to a first interface element in the target group if the current group is not the same as the target group, and wherein when the user navigation input is backward user navigation input I, input focus is shifted to an interface element previous in order in the current group if the current group is the same as the target group, or input focus is shifted to interface element last in order in the target group if the current group is not the same as the target group.* One of ordinary skill in the art would realize that Microsoft in view of Karp teaches a current group and target group as depicted in figure 3, such as when the user presses ALT the user is in the current group (parent group control UI elements) pending on the user desire of which menu group to activate the user can select groups using the forward navigation button (right arrow; on the keyboard) to go through the current group being that this is the target group however if this is not the target group of the user then pressing the sibling forward navigation button (down arrow; on the keyboard) will go through in a forward manner to next control elements in this target group. There is also provided additional controls for sibling control group (tab "forward navigation" and tab + shift "backward navigation). Just like the forward navigation the user is presented with backward navigation (left arrow and down arrow) for navigation is target groups of parents and sibling context groups. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Karp into Microsoft, because Karp discloses the same software product of Microsoft.

Additional references which exhibit what was commonly known in the art (http://www.delphi3000.com/articles/article_1667.asp?SK=), herein referred to as "Delphi", to support the Examiners assertion that group identifiers are commonly known and widely used in the art, especially in that of Windows XP and IE which is taught by Karp and Microsoft. As what is commonly known in the art and is explained throughout the entire reference of Karp is the navigation and use of Windows XP, in such that focus is brought upon a parent control ("Tools" figure 3 of "Microsoft Windows XP Operating System" screen shot reference handout) for navigation using predefined controls for navigation. As explained in Karp on pages 558, 560, 562 and 563 parent navigation controls are defined to navigate between parent controls (by using the left and right arrows (backward and forward navigation modifier keys; wherein when the user presses and activates a key (the operating system's listener/daemon (http://en.wikipedia.org/wiki/Daemon_(computer_software)) reads the input key unique identifier to be able to distinguish among all of the keys on a keyboard/input device) it modifies the display to render and update the display to show the navigation has moved either backward or forward pending on the particular control selected by the user) on the keyboard; only the parent controls are accessed by using these keys, no other controls are accessed (child controls in particular are inaccessible through the left and right arrow keys). The user can then use child navigation input controls (up and down arrows as discussed by Karp on pages 558, 560, 562 and 563 for navigation of child controls only, by inputting with up and down arrow keys (backward and forward navigation modifier keys; wherein when the user presses and activates a key (the operating system's listener/daemon reads the input key unique identifier to be able to distinguish among all of the keys on a keyboard/input device) it modifies the display to render and update the display to show the navigation has moved either backward or forward in a vertical list pending on the particular control selected by the user) only child navigation controls are selected. Thus focus is on a graphical container ("Tools"), which includes (when activated upon) a plurality of child controls associated with the parent control ("Tools") as what is commonly known in the art is that the containers (which are all objects ("Tool" menu button along with (e.g. File, Edit, View, Spelling and Grammar... Language, Word Count..., etc) of the graphical user interface (Windows Shell) (http://en.wikipedia.org/wiki/Container_(data_structure)#Graphic_Containers and http://en.wikipedia.org/wiki/Windows_shell) wherein the containers, objects, etc are all uniquely identified (as exhibited by Delphi). Thus the group associated with the parent control "Tools" are in a container (as rendered to the display (figure 3) with an identifier associated with that container (as exhibited by Delphi) thus forming a group, therefore "group identifier". Therefore when the user has input focus on a Parent or Child control list basic navigation using predetermined keys (up/down or left/right arrow of the keyboard) provide parent and sibling navigation input (wherein also could be described as backward and forward modifier key press (as noted above in the section of Operating System daemon's and to what is commonly known in the art of having unique identifiers to distinguish between components in a hardware system), which include a second group identifier key press and a first group identifier key press, respectively.

As for dependent claims 3-8 and 19-22, Microsoft teaches the product of claim 1 and corresponding method of claim 18, wherein:

- the user interface elements have associated text labels, and wherein the user interface elements associated with the group identifier are user interface elements having an associated text label with a first character that matches the group identifier (fig.5; “3 Main” and “3 Home”).
- a character matches a group identifier if both are the same character regardless of character case (fig. 5).
- a character matches a group identifier if both are the same character in the same case (fig. 5).
- group the user interface elements into groups based on the first character of the associated text label of the elements at application run time (fig.5; of course, those skilled in the art will appreciated that when the user activates the root node/ or parent node to display children nodes in the explorer window that the list is read from a file and then drawn to the screen, dynamically; hence the graphics of the menus were not there before hand.)
- group only the user interface elements in a current screen of the application into groups based on the first character of the associated text label (fig.5;

wherein it is appreciated that the list are user defined, the function "sort by name" if clear to sort/organize the list by first characters of a control).

As for independent claims 9 and 23, Microsoft teaches a computer program product and corresponding method, tangibly embodied in a computer-readable storage medium, the product comprising instructions operable to cause data processing apparatus to execute a method for navigating user interface elements on a display screen; the interface elements being arranged in order into user interface elements groups having assigned group identifier characters; and the interface elements indicating, on the display screen, an element currently having focus to receive user input; the method comprising: detecting a sequence of one or more user navigation input (fig.2, 3,5; wherein the user can use the operating system defined keyboard control keys, shortcuts, hot keys, etc...) (fig.5; wherein the user presses a navigation key and it is assigned to the location of the group identifier; for instance if the user presses "Alt-f" the group identifier for that key is the file context menu to which focus it shifted towards), each user navigation input being a forward user navigation input or a backward user navigation input, the forward user navigation input comprises a forward modifier key press combined with a key press of a first group identifier character, and the backward user navigation input comprises a backward modifier key press combined with a key press of a second group identifier character (fig.3; wherein the keyboard has defined functionality given by the operating system such as the left and right arrows and the tab and shift-tab controls, etc..); generating a navigation string from the sequence of one or more group identifier characters for the one or more user navigation input (fig.2 and 3; e.g. alt-tab and alt-x; of course, those skilled in the art will appreciate that a keyboard string (alt-x) can be used as defined by the operating system); and shifting input focus to a user interface element identified by the navigation string (fig.2 and 3; e.g. alt-tab and alt-x). However Microsoft does not expressly point out and describe *wherein, when the user navigation input is detected, determining a current group, that contains the interface element currently having input focus, and determining a target group that corresponds to the group identifier key press; wherein when the user navigation input is the forward user navigation input, input focus is shifted to an interface element next in order in the current group if the current group is the same as the target group, or input focus is shifted to a first interface element in the target group if the current group is not the same as the target group, and wherein when the user navigation input is backward user navigation input I, input focus is shifted to an interface element previous in order in the current group if the current group is the same as the target group, or input focus is shifted to interface element last in order in the target group if the current group is not the same as the target group.* One of ordinary skill in the art would realize that Microsoft in view of Karp teaches a current group and target group as depicted in figure 3, such as when the user presses ALT the user is in the current group (parent group control UI elements) pending on the user desire of which menu group to activate the user can select groups using the forward navigation button (right arrow; on the keyboard) to go through the current group being that this is the target group however if this is not the target group of the user then pressing the sibling forward navigation button (down arrow; on the keyboard) will go through in a forward manner to next control elements in this target group. There is also provided additional controls for sibling control group (tab "forward navigation" and tab + shift "backward navigation). Just like the forward navigation the user is presented with backward navigation (left arrow and down arrow) for navigation is target groups of parents and sibling context groups. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Karp into Microsoft, because Karp discloses the same software product of Microsoft.

As for dependent claims 10-12 and 24-26, Microsoft teaches the product of claim 9 and corresponding method of claim 23, wherein instructions to detect a sequence of one or more navigation key presses comprise instructions to:

- detect a sequence of forward user navigation input presses (fig.7), the sequence having a first navigation key press and a last navigation key press (fig.3, left and right arrows, etc); initialize the navigation string when the first navigation key press is detected (of course those skilled in the art will appreciate that when the user presses a key sequence/ string of keys the operating system listener for that program will communicate and act on the keys pressed; start a time out interval with each forward user navigation input press; and determine the last navigation key press as the key press after which no forward user navigation input presses are detected within the time out interval.

- detect a sequence of backward user navigation input presses, the sequence having a first navigation key press and a last navigation key press; initialize the navigation string when the first navigation key press is detected; start a time out interval with each backward user navigation input press (associated with the listener of the operating system); and determine the last navigation key press as the key press after which no backward user navigation input presses are detected within the time out interval (note the above analysis of forward navigation).
- shift input focus to a next user interface element having a text label starting with the same characters as the characters in the navigation string, if the navigation key is a forward user navigation input; and shift input focus to a previous user interface element having a text label starting with the same characters as the characters in the navigation string, if the navigation key is a backward user navigation input (fig.2,3 and 5; wherein focus is being shown).

As for independent claims 13 and 27, Microsoft teaches a computer program product and corresponding method, tangibly embodied in an computer-readable storage medium, the product comprising instructions operable to cause data processing apparatus to: execute a method for navigating user interface elements on a display screen; the interface elements being arranged in order into user interface elements groups having assigned group identifier characters; and the interface elements indicating, on the display screen, an element currently having focus to receive user input; the method comprising: detecting an ensemble of sequential user activation inputs, each user activation input comprising a character (note the analysis of claims 1 and 9), thereby detecting a sequence of characters; each user activation input comprising one of a forward user activation input or a backward user activation input, the forward user activation input comprises a forward activation modifier key press combined with key press of a first group identifier character and the backward user activation input comprises a backward activation modifier key press combined with a key press of a second group identifier character key; identifying a matching activation user interface element by finding an activation user interface element having a label matching the sequence of characters; and perform an action associated with the matching activation user interface element (note the analysis of claims 1 and 9; e.g. the user presses alt-x, wherein x is related to a character of a control so the user can scroll through the parent nodes of selection based on character association). However Microsoft does not expressly point out and describe *wherein, when the user activation input is detected, determining a current group, that contains the interface element currently having input focus, and determining a target group that corresponds to the group identifier key press; wherein when the user activation input is the forward user navigation input, input focus is shifted to an interface element next in order in the current group if the current group is the same as the target group, or input focus is shifted to a first* *interface element in the target group if the current group is not the same as the target group, and wherein when the user activation input is backward user navigation input I, input focus is shifted to an interface element previous in order in the current group if the current group is the same as the target group, or input focus is shifted to interface element last in order in the target group if the current group is not the same as the target group.* One of ordinary skill in the art would realize that Microsoft in view of Karp teaches a current group and target group as depicted in figure 3, such as when the user presses ALT the user is in the current group (parent group control UI elements) pending on the user desire of which menu group to activate the user can select groups using the forward navigation button (right arrow; on the keyboard) to go through the current group being that this is the target group however if this is not the target group of the user then pressing the sibling forward navigation button (down arrow; on the keyboard) will go through in a forward manner to next control elements in this target group. There is also provided additional controls for sibling control group (tab "forward navigation" and tab + shift "backward navigation). Just like the forward navigation the user is presented with backward navigation (left arrow and down arrow) for navigation is target groups of parents and sibling context groups. It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Karp into Microsoft, because Karp discloses the same software product of Microsoft. (Also note the analysis of claim 1 above).

As for dependent claims 14-17 and 28-31, Microsoft teaches the product of claim 13 and corresponding method of claim 27, wherein instructions to detect an ensemble comprise instructions to:

- detect a sequence of one or more characters that uniquely identifies an activation user interface element (note analysis of claim 13, 1 and 9; "alt-x", etc...).
- the sequence of one or more characters is a sequence of identical group identifiers (note the analysis of claims 13, 1 and 9; "alt-x", etc).
- detect one or more sequential activation key presses entered by a user within a time threshold (note the analysis of claims 1,9 and 13; wherein the operating system has a listener for the explorer application to listen for user imputer from peripheral devices.)
- the pressing and releasing of an activation modifier key delimits the activation key presses in the ensemble (of course those skilled in the art will appreciate that if the user presses a key command on the keyboard that the listener will send command with the appropriate action associated with the appropriate control).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and

> the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 3-7,20 and 21 are rejected under 35 U.S.C. 103(a) as unpatentable by Microsoft in view of Karp or, in the alternative, under 35 U.S.C. 103(a) as obvious over Microsoft in view of Karp in further view of Benhase et al (US 2004/0243616). Microsoft in view of Karp teaches as mentioned above, but in more support Benhase teaches:

- the user interface elements have associated text labels, and wherein the user interface elements associated with the group identifier are user interface elements having an associated text label with a first character that matches the group identifier (fig.3;par.36).
- a character matches a group identifier if both are the same character regardless of character case (fig.3;par.36).

- a character matches a group identifier if both are the same character in the same case (fig.3;par.36).
- group the user interface elements into groups based on the first character of the associated text label of the elements at application run time (fig.3;par.36).
- group only the user interface elements in a current screen of the application into groups based on the first character of the associated text label (fig.3;par.36).

It would have been obvious to one of ordinary skill in the art at the time of the invention to combine the product/ method of Microsoft into the product/ method of Benhase. This is true because the windows explorer a tree based file directory can be displayed adjacent to a list or table of files and associated information on the computer monitor. For example, the tree may indicate various directories and subdirectories (controls, links to) arranged in an expandable and collapsible format (par.4, lines 4-10).

(Note:) It is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006,1009, 158 USPQ 275, 277 (CCPA 1968)).

Response to Arguments

Applicant's arguments filed 10/17/2007 have been fully considered but they are not persuasive.

A1. Applicant argues the cited references either alone or in combination does not teach "the forward user navigation input comprises a forward modifier key press combined with a key press of a first group identifier character, and the backward user navigation input comprises a backward modifier key press combined with a key press of a second group identifier character".

R1. After careful review of the amended limitations along with the remaining limitations previous presented the Examiner does not agree with the Applicant in the aspect that Windows XP and Microsoft Internet Explorer (IE) (wherein IE is apart of Windows XP and takes on the implemented functionality and design of Windows XP that is discussed by Karp) as commonly known in the art exhibits and makes use of identifiers. The Examiner provides additional references which exhibit what was commonly known in the art (http://www.delphi3000.com/articles/article_1667.asp?SK=), herein referred to as "Delphi", to support the Examiners assertion that group identifiers are commonly known and widely used in the art, especially in that of Windows XP and IE which is taught by Karp and Microsoft. As what is commonly known in the art and is explained throughout the entire reference of Karp is the navigation and use of Windows XP, in such that focus is brought upon a parent control ("Tools" figure 3 of "Microsoft Windows XP Operating System" screen shot reference handout) for navigation using predefined controls for navigation. As explained in Karp on pages 558, 560, 562 and 563 parent navigation controls are defined to navigate between parent controls (by using the left and right arrows (backward and forward navigation modifier keys; wherein when the user presses and activates a key (the operating system's listener/daemon (http://en.wikipedia.org/wiki/Daemon_(computer_software)) reads the input key unique identifier to be able to distinguish among all of the keys on a keyboard/input device) it modifies the display to render and update the display to show the navigation has moved either backward or forward pending on the particular control selected by the user) on the keyboard; only the parent controls are accessed by using these keys, no other controls are accessed (child controls in particular are inaccessible through the left and right arrow keys). The user can then use child navigation input controls (up and down arrows as discussed by Karp on pages 558, 560, 562 and 563 for navigation of child controls only, by inputting with up and down arrow keys (backward and forward navigation modifier keys; wherein when the user presses and activates a key (the operating system's listener/daemon reads the input key unique identifier to be able to distinguish among all of the keys on a keyboard/input device) it modifies the display to render and update the display to show the navigation has moved either backward or forward in a vertical list pending on the particular control selected by the user) only child navigation controls are selected. Thus focus is on a graphical container ("Tools"), which includes (when activated upon) a plurality of child controls associated with the parent control ("Tools") as what is commonly known in the art is that the containers (which are all objects ("Tool" menu button along with (e.g. File, Edit, View, Spelling and Grammar... Language, Word Count..., etc) of the graphical user interface (Windows Shell) (http://en.wikipedia.org/wiki/Container_(data_structure)#Graphic_Containers and http://en.wikipedia.org/wiki/Windows_shell) wherein the containers, objects, etc are all uniquely identified (as exhibited by Delphi). Thus the group associated with the parent control "Tools" are in a container (as rendered to the display (figure 3) with an identifier associated with that container (as exhibited by Delphi) thus forming a group, therefore "group identifier". Therefore when the user has input focus on a Parent or Child control list basic navigation using predetermined keys (up/down or left/right arrow of the keyboard) provide parent and sibling navigation input (wherein also could be described as backward and forward modifier key press (as noted above in the section of Operating System daemon's and to what is commonly known in the art of having unique identifiers to distinguish between components in a hardware system), which include a second group identifier key press and a first group identifier key press, respectively. The Examiner believes with this understanding to the Applicant that the Applicant can know agree that the Examiner has established a prima facie case of obviousness, wherein it is believed by the Examiner that the cited references used in this rejection teach each and every limitation of the immediate application.

A2. Applicant argues claims 3-7,20 and 21 with respect to the similar limitation in argument A1 above.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Prior art cited is related to navigation techniques.

Inquires

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nicholas Augustine whose telephone number is 571-270-1056. The examiner can normally be reached on Monday - Friday: 7:30- 5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

N. Augustine
January 4, 2008

Nicholas Augustine
Examiner
AU: 2179

BA HUYNH
PRIMARY EXAMINER